

Challenging common bolus advisor for self-monitoring type-I diabetes patients using Reinforcement Learning

Frédéric Logé

frederic.logemunerel@gmail.com
Air Liquide R&D, Jouy-en-Josas, FR
CMAP, Polytechnique, Institut
Polytechnique de Paris, Palaiseau, FR

Erwan Le Pennec

CMAP, Polytechnique, Institut
Polytechnique de Paris, Palaiseau, FR
XPop, Inria Saclay, FR

Habiboulaye

Amadou-Boubacar
Air Liquide R&D, Jouy-en-Josas, FR

ABSTRACT

Patients with diabetes who are self-monitoring have to decide right before each meal how much insulin they should take. A standard bolus advisor exists, but has never actually been proven to be optimal in any sense. We challenged this rule applying Reinforcement Learning techniques on data simulated with T1DM, an FDA-approved simulator developed by [3] modeling the gluco-insulin interaction. Results show that the optimal bolus rule is fairly different from the standard bolus advisor, and if followed can actually avoid hypoglycemia episodes.

CCS CONCEPTS

• **Applied computing** → **Consumer health**; • **Theory of computation** → **Reinforcement learning**.

1 INTRODUCTION

Diabetes & bolus advisor. Diabetes is a major disease which requires, amongst many things, patients to keep their blood glucose (BG) levels in check. To get an idea of commonly accepted levels: if the BG is lower than 70 mg/dL, the patient is in hypoglycemia, if higher than 180 mg/dL, the patient is in hyperglycemia and otherwise in normoglycemia, the ideal is to be around 112.5 mg/dL. When BG is too low, the patient must take carbohydrates (CHO) to compensate. When it is too high, the patient must take external insulin in order to compensate low pancreatic activity. Insulin intake can be done via an insulin pump which injects continuously varying amounts throughout the day. Most patients however are still injecting punctually, most of the time prior to their meals in order to prepare for the future CHO intake and avoid hyperglycemia.

A standard rule to follow for pre-meal self-injections of insulin is the following piecewise linear formula : for a given measure of blood glucose BG and an intake of carbohydrates CHO, one should take the following bolus¹ quantity :

$$\text{bolus}(\text{CHO}, \text{BG}) = \frac{\text{CHO}}{\text{CIR}} + \frac{\max\{\text{BG} - \text{BG}_{\text{target}}, 0\}}{\text{CF}} \quad (1)$$

in order to reach the BG target level $\text{BG}_{\text{target}}$. In this rule, commonly known as bolus advisor, coefficients CIR and CF correspond to the carbohydrate-to-insulin ratio and the correction factor respectively, both being specific to each individual and can be evaluated by medical tests and usually are fluctuant during the day. The first ratio in the rule compensates for CHO intake and the second ratio compensates for excessively high BG levels.

¹Bolus insulin corresponds to rapid insulin injections whereas basal insulin corresponds to long-term insulin.

Related works. A lot of the research around blood glucose management for diabetes focuses on the artificial pancreas, so the case where the patient is equipped with an insulin pump. The interested reader can find an extensive review here [1]. For self-monitoring, [6] worked on the best delivery of insulin drugs to facilitate BG management. Based on a complex diabetes simulator, the authors of [2] and [7] worked on learning adaptively coefficients (CIR, CF) which is crucial for deployment.

Objective. Although the bolus advisor makes sense, there has not been any study of its performance in its use for patients who are self-monitoring. In this study, we rely on simulated data for type-I diabetes patients and a model-free approach to actually learn the optimal bolus function, tailoring it to meal plans. Using the vocabulary of Reinforcement Learning, we show how to learn non parametric *policies*, the function giving insulin doses based on BG readings and auxiliary information, that outperforms the standard advisor for a given meal pattern.

2 METHODOLOGY

The problem will be framed as a Markov Decision Process, as presented in the first subsection. The resolution technique, a model-free Reinforcement Learning approach, will be presented next. Finally, we will talk about the use of the simulator available to generate the necessary data for our experiments.

2.1 Markov Decision Process

To model the insulin intake process we consider a Markov Decision Process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbf{R}, \mathbf{T})$ where $(\mathcal{S}, \mathcal{A})$ denote the state and action spaces, \mathbf{R} is the reward function and \mathbf{T} is the transition function.

Time steps are set on meals, which follow a strict regular plan. The state variable S_t contains the meal identifier ID_meal_t , a categorical variable, and the BG reading BG_t registered before the meal. The action variable A_t contains the insulin dose INS_t taken before the meal. Based on the meal plan, we also know the amount of CHO associated with the meal CHO_t , which should simply be a function of ID_meal_t . The reward R_t received is a function of BG_t , as represented in figure 1 which is indicative of the wellness of the BG profile: the closer it is to 112.5 mg/dL the better.

Let us call *policy* any function π which maps \mathcal{S} to \mathcal{A} i.e. any function which to a couple $(\text{ID_meal}, \text{BG})$ assigns an insulin recommendation INS . Writing Π the set of policies, we ought to find π^* , the policy maximizing cumulative rewards:

$$\pi^* = \arg \max_{\pi \in \Pi} \mathbb{E}_{\mathcal{M}, \pi} \left[\sum_{t=0}^{+\infty} \gamma^t R_t \right] \quad (2)$$

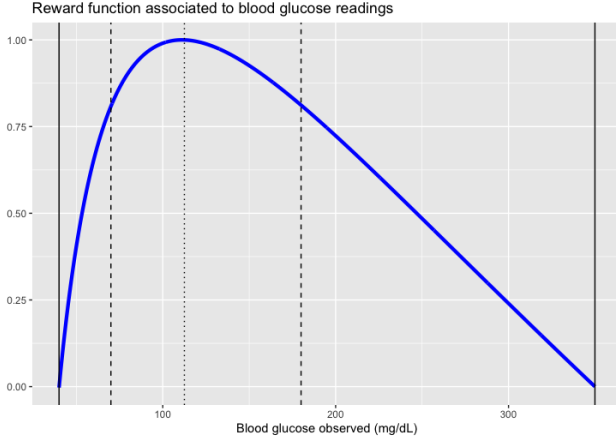


Figure 1: Reward function for BG levels inspired by [4], where the authors proposed to build a function centered at BG level 112.5 mg/dL and symmetric through reference levels, shown with vertical lines.

where discount factor $\gamma \in (0, 1)$ is chosen by the user and indicates how much long-term rewards matter. Note that the standard bolus advisor from equation 1 belongs to Π .

2.2 Q-Learning with function approximation

For any policy $\pi \in \Pi$ and any couple $(s, a) \in \mathcal{S} \times \mathcal{A}$, consider the state-action value functions

$$Q_\pi(s, a) \doteq \mathbb{E}_{\mathcal{M}, \pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t | S_0 = s, A_0 = a \right]. \quad (3)$$

Because our state and action spaces are continuous, a common approach to solve the optimization problem is to parameterize the state-action value functions and find the optimal parameters instead. In our case, we will consider the following linear function approximation of $Q(s, a)$ for $s = (\text{ID_meal}, \text{BG})$, $a = \text{INS}$:

$$Q_{\bar{\alpha}}(s, a) \doteq \sum_{b=1}^B \sum_{b'=1}^B \alpha_{(b, b')}^{(\text{ID_meal})} \phi_b^{\text{BG}}(\text{BG}) \phi_{b'}^{\text{INS}}(\text{INS}) \quad (4)$$

with vector parameter $\bar{\alpha}$. The set of functions ϕ_b^{BG} and $\phi_{b'}^{\text{INS}}$ are taken as radial basis functions, centered uniformly on their respective grids. The scaling was made in order to have a certain percentage p of overlap amongst neighbour functions. The number of functions B handles the granularity of the grid.

Linear approximation scheme is widely used in the RL literature, some details can be found in chapter 9 of [8].

Note that in our context we face the *deadly triad* issue [8] since we jointly apply bootstrap, function approximation and off-policy learning. In order to alleviate this issue, we use a frozen set $\bar{\alpha}^{(\text{frozen})}$ to stabilize target value and apply replay memory to break temporal dependencies between transitions. This strategy has shown good results for instance in [5] where Deep Neural Networks were used as function approximators for Atari game controllers.

2.3 Gluco-Insulin interactions Simulator

T1DM simulator (2008 version, accessed throughout Python implementation [9]) is described in detail in [3]. It models the glucose and insulin dynamics taking into account

- for glucose : endogenous production, rate of appearance, utilization and renal extraction
- for insulin : rate of appearance from the subcutaneous tissue and insulin degradation.

The simulator takes into account physiology, a given meal scenario and any insulin basal/bolus policy. It also provides good base values for Carbohydrate-to-Insulin Ratio, Correction Factor and Basal Rate for a given physiology.

Although it is built for continuous glucose monitoring, we can use it to generate data in the self-monitoring setting. Using the simulator, the data generated is a set alike

$$(\text{BG}_t, \text{CHO}_t, \text{INS}_t)_{t=1}^T \quad (5)$$

with a time step of 3 minutes and where the CHO intake depends on the user-specified meal plan and the insulin level corresponds to the bolus insulin which is specified by the user. In our case, the insulin component was sampled uniformly on a proper range of values. The data is then summarized to pre-meal observations as

$$(\text{ID_meal}_t, \text{BG}_t, \text{CHO}_t, \text{INS}_t, R_t)_{t \in \text{meal_times}}. \quad (6)$$

3 NUMERICAL EXPERIMENTS

Experiment outline. We proceeded as follows :

- Step 0 : choose meal scenario of interest and virtual patient
- Step 1 : approximate optimal coefficients for the baseline bolus advisor via grid search (policy π_0)
- Step 2 : simulate a large amount of data following this meal scenario and explore randomly the insulin intakes within acceptable range, set using the bolus advisor with lower and upper bounds on coefficients (CIR, CF)
- Step 3 : approximate optimal bolus from simulated data with RL approach (policy $\pi_{\bar{\alpha}^*}$)
- Step 4 : simulate validation data following policies π_0 and $\pi_{\bar{\alpha}^*}$, compare policy functions and daily glycemic profiles obtained.

Meal scenario. We chose the following meal scenario: at 6am the person takes 50 grams of CHO for breakfast. For lunch, at 12am, the person takes 60 grams of CHO, followed by a snack of 15 grams at 3pm. At 8pm, for dinner, the person takes 80 grams of CHO.

Simulation details. The procedure above was applied to adults #1, #2 and #3 from simulator T1DM. We simulated for each of them around 8 years of data which amounts to 11680 observations (meals) each. We based our grid search for π_0 from the following bounds : CIR $\in [3; 30]$, CF $\in [0.4; 2.8]$ and $\text{BG}_{\text{target}} \in [100 \text{ mg/dL}; 150 \text{ mg/dL}]$. For the Q-Learning algorithm, we set $B = 8$, giving an 8 by 8 grid of the (BG, INS) plane and we set $p = 0.2$ which gave, from our qualitative evaluation, satisfying result.

Based on the initial data generated, the policies were tested for the same virtual patients (separately, each adult has his/her policy) during 45 days, which is sufficient since the BG enters a stationary state because meal scenario was set deterministic.

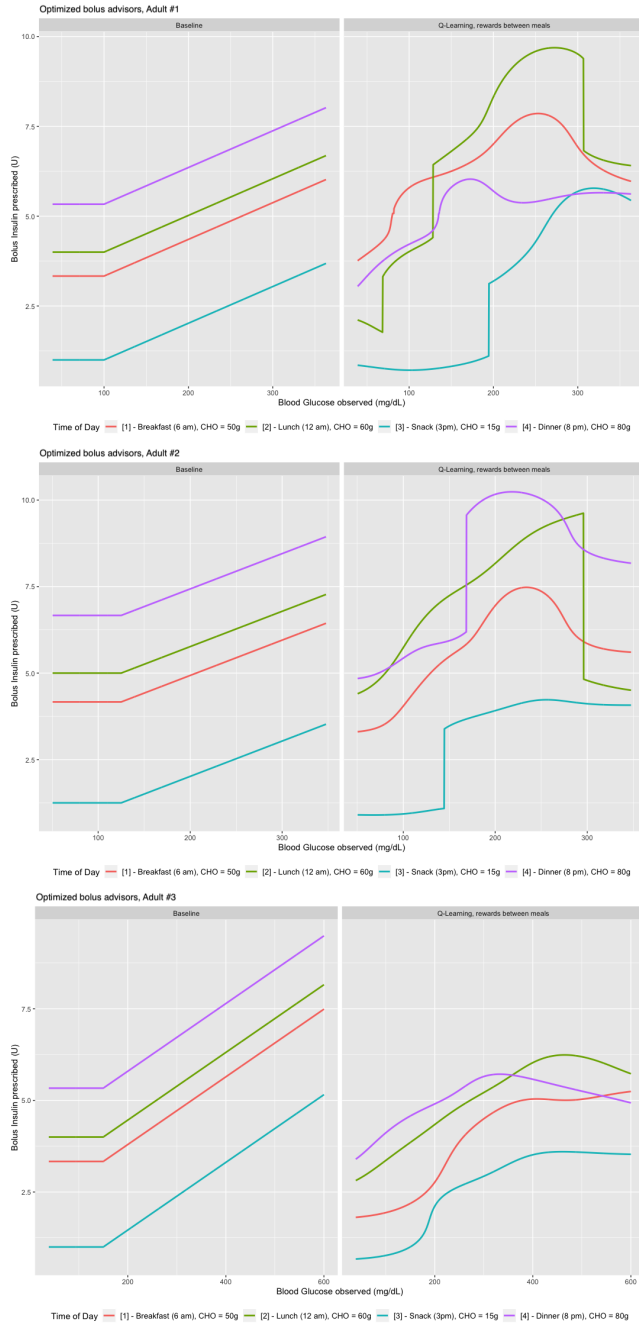


Figure 2: Bolus advisor π_0 (left) and $\pi_{\tilde{\alpha}^*}$ found by Q-Learning with function approximation (right) for each virtual patient (sorted by increasing number). One can observe essentially two things from those graphs : first, it looks as if patient #1 and #2 are closer to each other than to patient #3 in terms of physiology otherwise we would not find the result we did. However the hope to get out a new generic rule seems difficult in this situation. Second, and this is what shows the algorithm took into account sequentiality of the meals and injections : the meal with most carbs does not necessarily require most insulin, which is a fundamental difference from the baseline advisor.

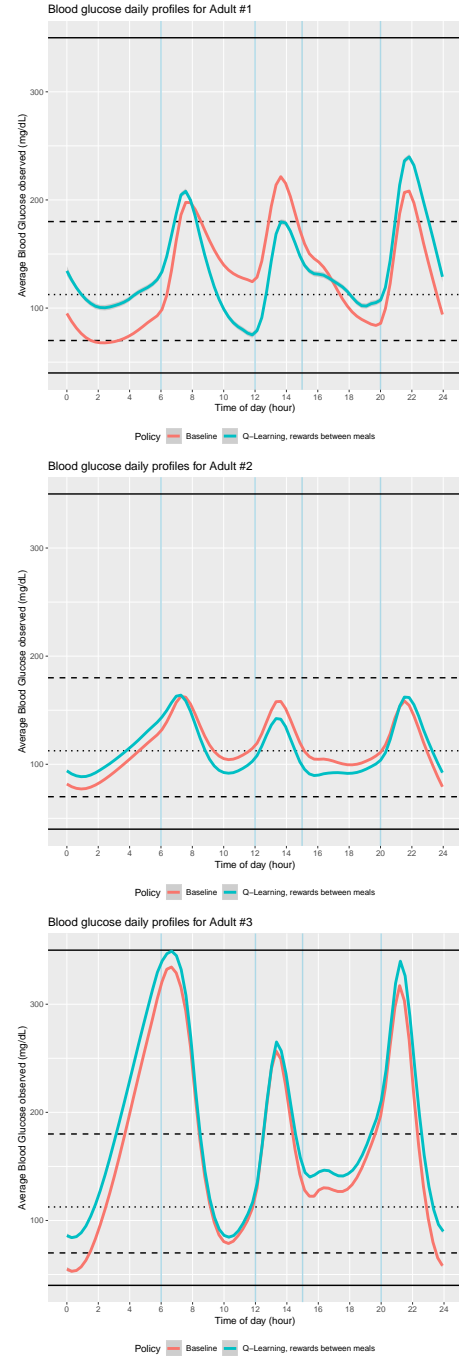


Figure 3: Average BG profiles for the three virtual patients (sorted increasingly by number) for each policy applied, π_0 in red and $\pi_{\tilde{\alpha}^*}$ in light blue. For all adults, the algorithm basically learned to reduce insulin intake at last meal in order to reduce nightly hypoglycemia at the cost of often minor hyperglycemia. The changes are minor in the case of Adult #2, but in Adult #3 and #1 it centers the BG levels around target 112.5 mg/dL.

Result disposition. In table 1, we included the distribution of BG levels on test data, essentially showing that for adults #1 and #3, policy $\pi_{\bar{\alpha}}$ lessens the BG variations outside normoglycemic range compared to π_0 . In figure 2 we present jointly the two policies for each patient, in the left column the standard bolus advisor π_0 and in the right column the RL-optimized policy $\pi_{\bar{\alpha}}$. In figure 3 we compare the BG average daily profiles obtained between for the two policies ($\pi_{\bar{\alpha}}$ is in blue).

BG distribution	Adult #1		Adult #2		Adult #3	
	π_0	$\pi_{\bar{\alpha}}$	π_0	$\pi_{\bar{\alpha}}$	π_0	$\pi_{\bar{\alpha}}$
[40 mg/dL, 70 mg/dL)	.07	.00	.00	.00	.08	.00
[70 mg/dL, 112.5 mg/dL)	.35	.37	.53	.58	.18	.20
[112.5 mg/dL, 180 mg/dL)	.39	.47	.47	.42	.35	.35
[180 mg/dL, 350 mg/dL)	.19	.16	.00	.00	.35	.39
[350 mg/dL, 600 mg/dL]	.00	.00	.00	.00	.04	.06

Table 1: BG distribution for both policies and each patient. Lecture key: for adult #1, following π_0 , 7% of BG readings fell within interval [40 mg/dL, 70 mg/dL).

Policies found, figure 2. Policies $\pi_{\bar{\alpha}}$ are quite different in form compared to π_0 and their form varies with patients. Also, for a given BG level, the recommended bolus does not necessarily increase with CHO amount ; we tend to give more insulin at lunch rather than at dinner, which impacts greatly the BG profile. Typically the patient will be less at risk from nocturnal hypoglycemia which is a well-known issue in the endocrinology community. The policy we proposed properly took into consideration the meal scenario in order to look at future time steps and prevent complications. One may also note that the bolus advisors calibrated sometimes recommend decreasing amounts of insulin with increasing BG level: a great example of that is the lunch recommendations for adults #1 and #2 : when BG goes over 300 mg/dL the recommended insulin level drops. This is due to the scarcity of the data collected within this area so it should be considered as a numerical artefact.

Daily BG profile comparison, figure 3. Let us focus on Adult #1, whose data is represented in the upper graph. The baseline retained induces low BG during the night (ranging in 70 mg/dL to 100 mg/dL) whilst it stays quite higher than 112.5 mg/dL from 6am to 3pm. The policy we found (in blue) eliminates night hypoglycemia, traded-off for a two hours period in mild hyperglycemia (<250 mg/dL) during dinner and a more smoothed-out afternoon. For Adult #2, the two policies seem quite equivalent. For Adult #3, we trade-off the night hypoglycemic section by hyperglycemia on the three main meal times.

4 DISCUSSION

Contributions. In this work we challenge a commonly used bolus advisor by relying on virtual patients data, using T1DM simulator [3] and a model-free approach from Reinforcement Learning. The policies found bring BG levels closer to reference level of 112.5 mg/dL and in two out of the three adults managed to avoid nocturnal hypoglycemia. These first results attest that a proper bolus advisor does need to be tailored to a patient’s physiology

(as expressed in the standard bolus advisor through personalized coefficients (CIR, CF)), but they should also be tailored to meal and activity plans, which is what we propose here. The code developed for those experiments is available at https://github.com/FredericLoge/T1DM_qlearning.

Future work. We are considering four developments of this work. First, we would like to extend the Reinforcement Learning approach using neural networks as function approximators to take into account historical data properly. Second, instead of optimizing only bolus insulin intake, we could jointly optimize meal and potential activity plans so as to not stress the body with high BG variations. Third, we would like to try different reward functions and compare qualitatively obtained policies. Last but not least, we would like to investigate how quickly we could learn the optimal policies we found here, which did require years of simulated data with randomized insulin intakes, which is obviously unfeasible in real-life.

ACKNOWLEDGMENTS

We would like to thank our colleague at Air Liquide, Fouad Megherbi, MD, Medical Director, for his input on this topic and medical references he provided us with to enrich the study.

CONFLICT OF INTEREST DISCLAIMER

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

- [1] Melanie K Bothe, Luke Dickens, Katrin Reichel, Arn Tellmann, Björn Ellger, Martin Westphal, and Ahmed A Faisal. 2013. The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas. *Expert review of medical devices* 10, 5 (2013), 661–673.
- [2] Elena Daskalaki, Peter Diem, and Stavroula G Mougiakakou. 2013. An Actor-Critic based controller for glucose regulation in type 1 diabetes. *Computer methods and programs in biomedicine* 109, 2 (2013), 116–125.
- [3] Boris P Kovatchev, Marc Breton, Chiara Dalla Man, and Claudio Cobelli. 2009. In silico preclinical trials: a proof of concept in closed-loop control of type 1 diabetes.
- [4] Boris P Kovatchev, Martin Straume, Daniel J Cox, and Leon S Farhy. 2000. Risk analysis of blood glucose data: a quantitative approach to optimizing the control of insulin dependent diabetes. *Computational and Mathematical Methods in Medicine* 3, 1 (2000), 1–10.
- [5] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).
- [6] Phuong D Ngo, Susan Wei, Anna Holubová, Jan Muzik, and Fred Godtliessen. 2018. Control of blood glucose for type-1 diabetes by using reinforcement learning with feedforward algorithm. *Computational and mathematical methods in medicine* 2018 (2018).
- [7] Qingnan Sun, Marko V Jankovic, João Budzinski, Brett Moore, Peter Diem, Christoph Stettler, and Stavroula G Mougiakakou. 2018. A dual mode adaptive basal-bolus advisor based on reinforcement learning. *IEEE journal of biomedical and health informatics* 23, 6 (2018), 2633–2641.
- [8] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [9] Jinyu Xie. 2018. Simglucose v0.2.1. <https://github.com/jxx123/simglucose>. [Online; accessed 05-December-2019].